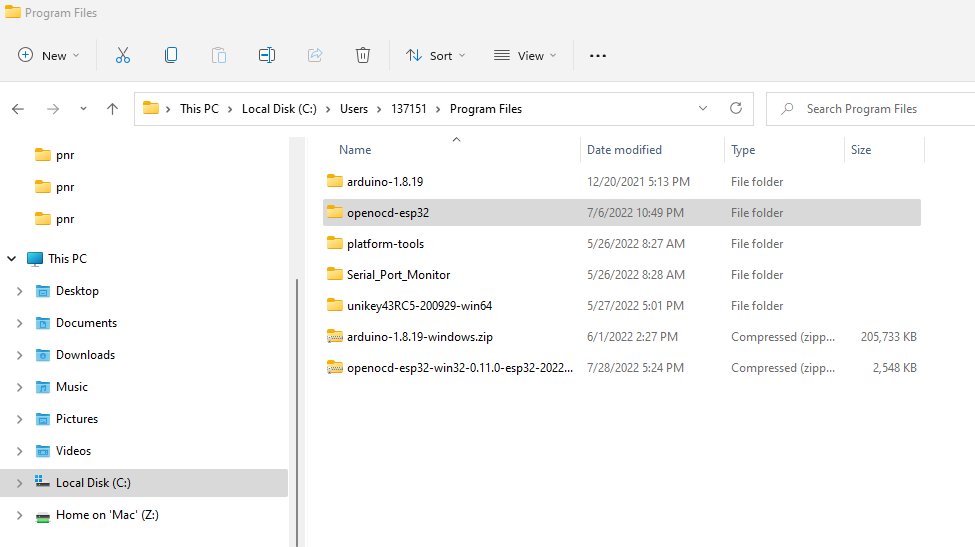
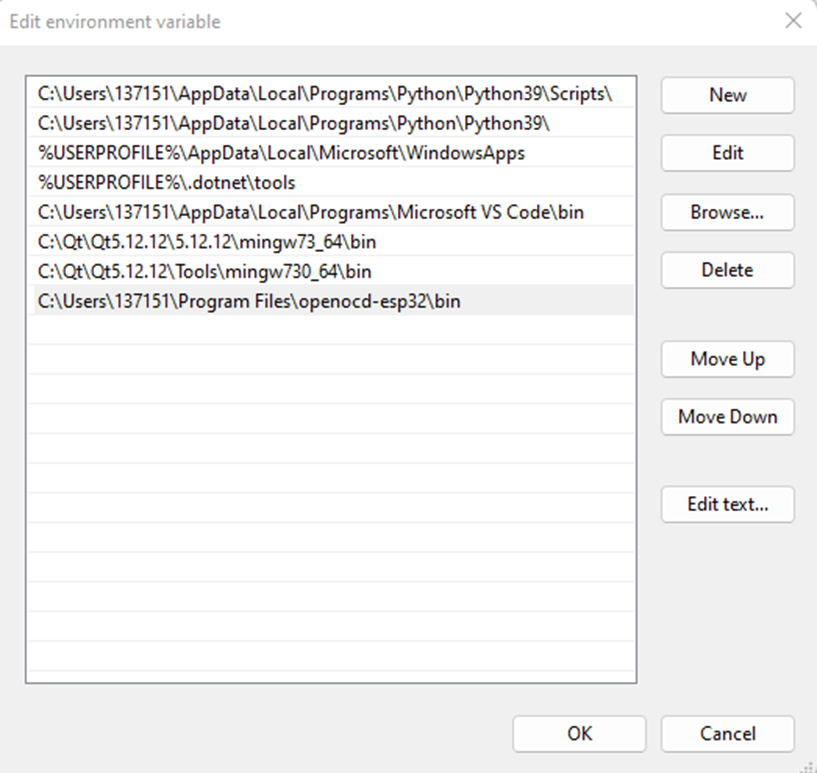
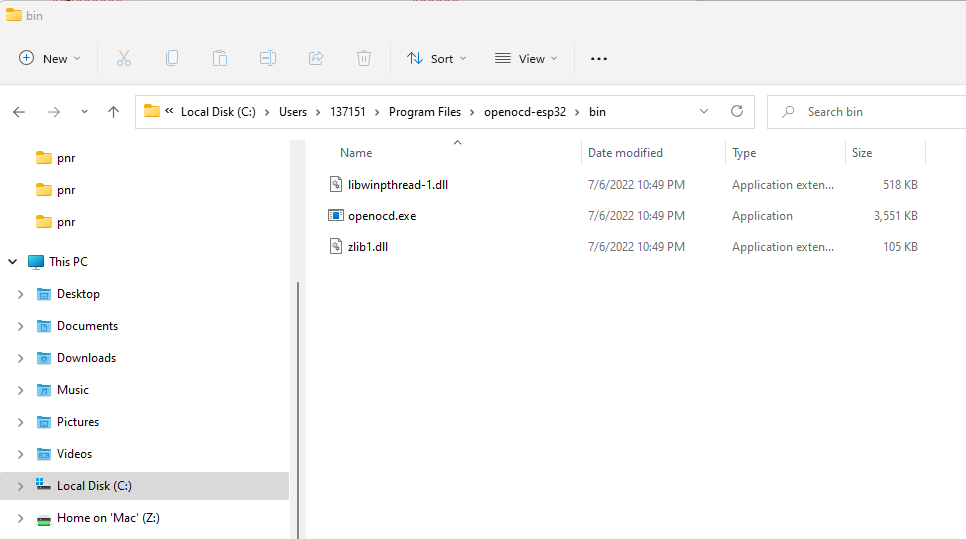
**1. Install OpenOCD for ESP32**

* Reference:
  + https://docs.espressif.com/projects/esp-idf/en/v3.3/api-guides/jtag-debugging/setup-openocd-windows.html
* Download OpenOCD for Windows
  + OpenOCD for Windows is available for download from Github: <https://github.com/espressif/openocd-esp32/releases>
* Extract OpenOCD package. Add OpenOCD to path environment.





* Check version of OpenOCD
  + openocd.exe –version

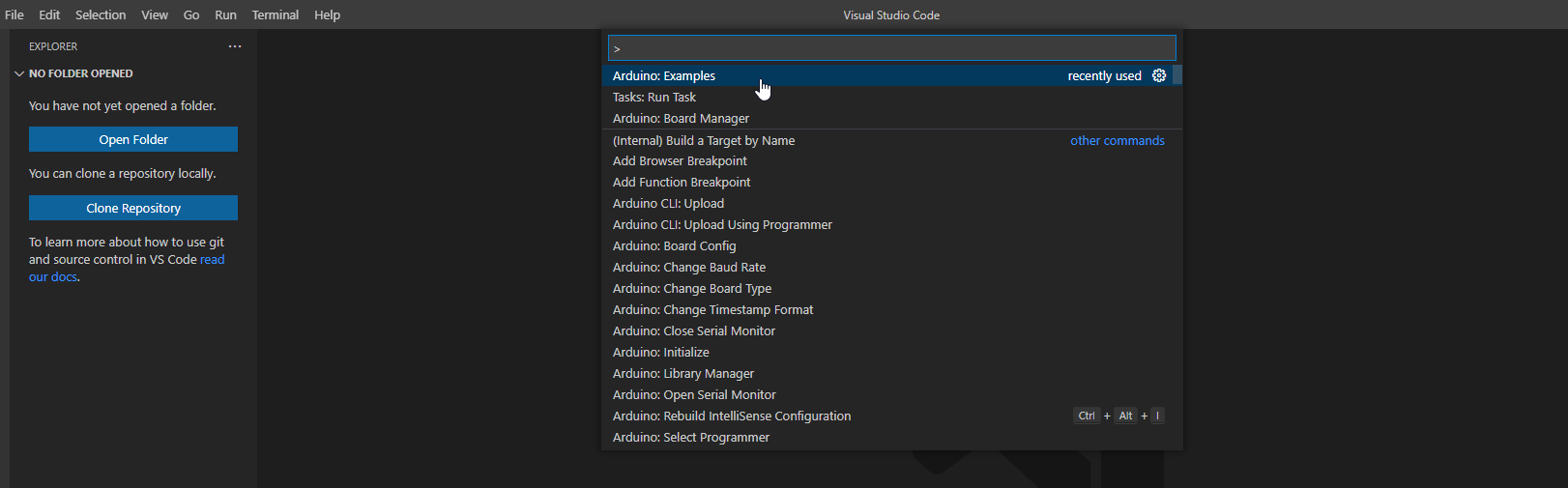


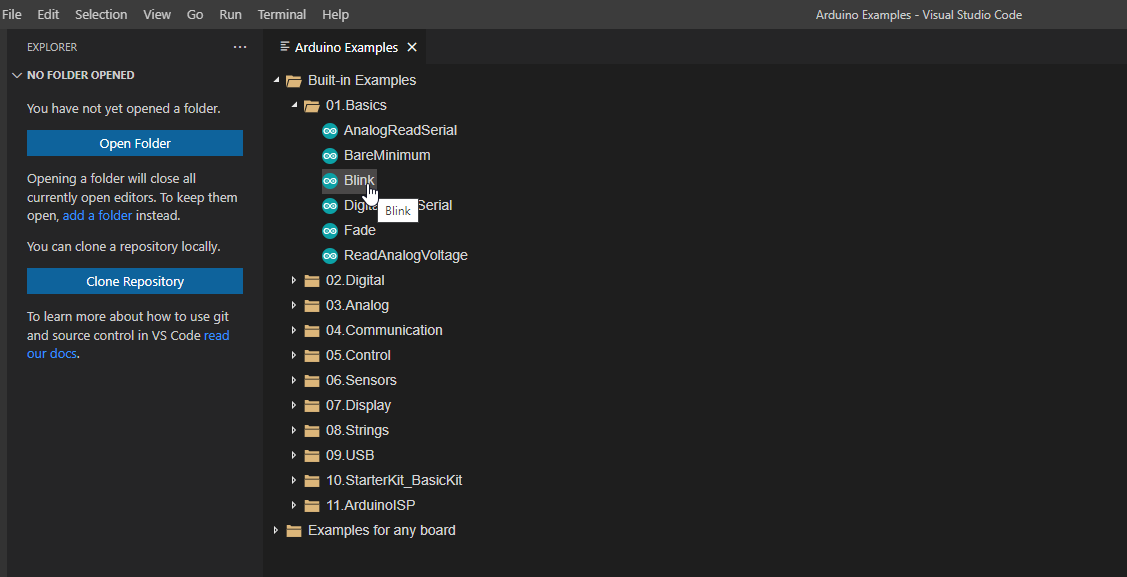
**2. OpenOCD for ESP32 use FT2232 minimodule**

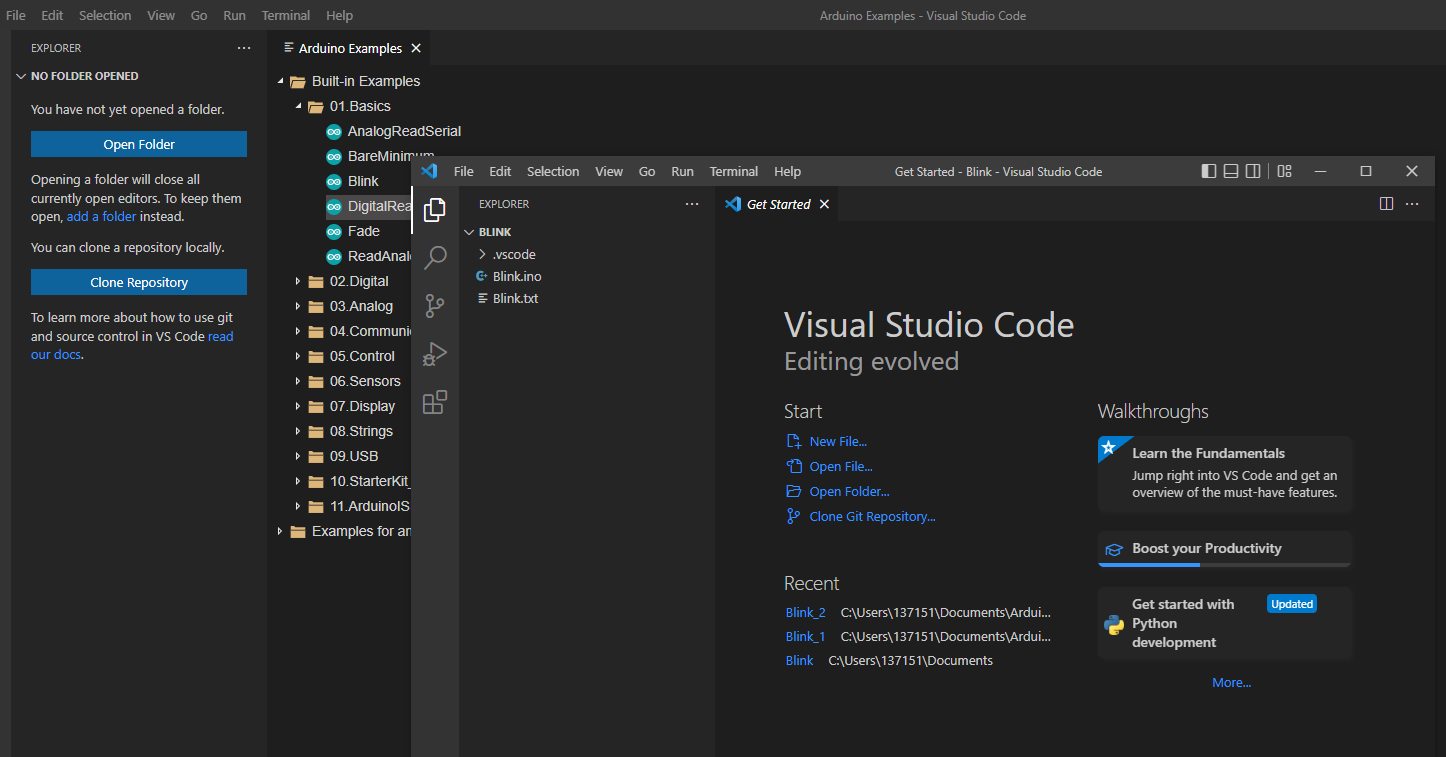
* Reference
  + <https://mcuoneclipse.com/2019/10/20/jtag-debugging-the-esp32-with-ft2232-and-openocd/>
  + <https://docs.espressif.com/projects/esp-idf/en/v3.3/api-guides/jtag-debugging/index.html#jtag-debugging-configuring-esp32-target>

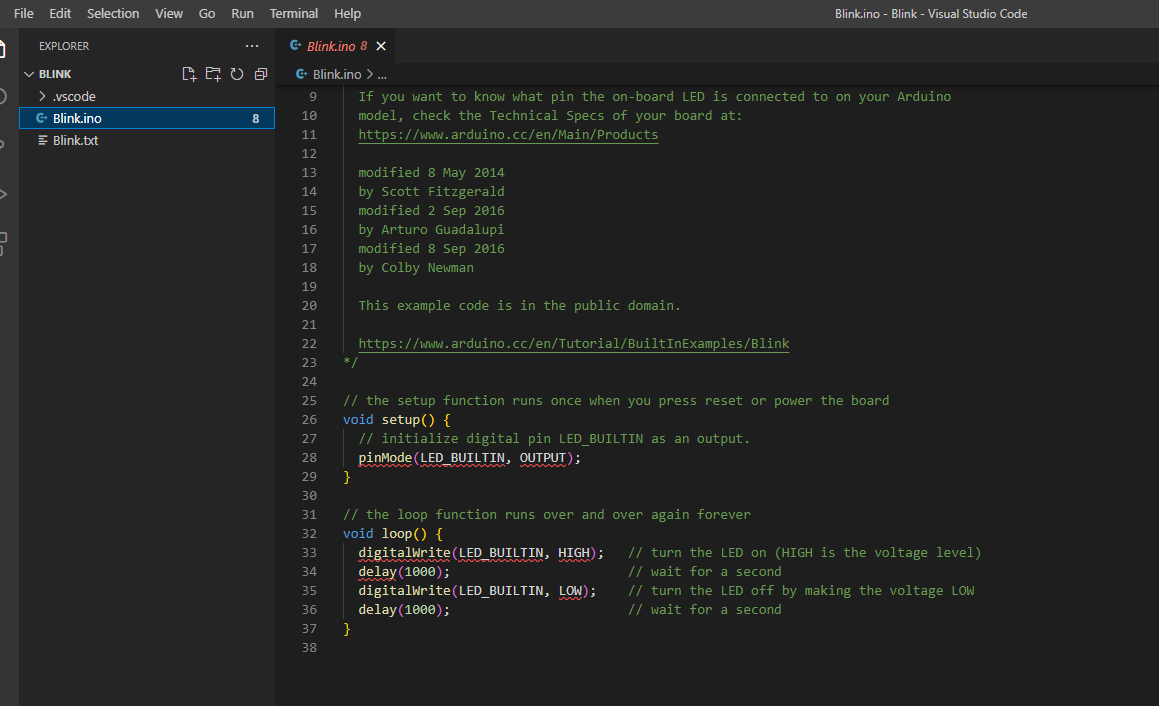
**3. Debugging ESP32 by JTAG with OpenOCD + GDB on Visual Studio Code for Arduino**

* Install Arduino Extension on Visual Studio Code
* Install ESP32 package
* Create Arduino Blink project from example
  + F1 => Typing “Arduino: Examples”

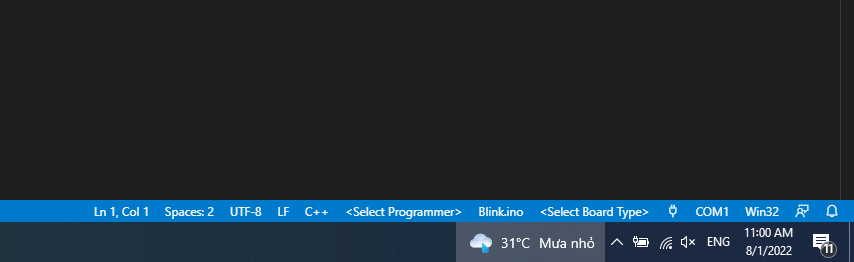
****

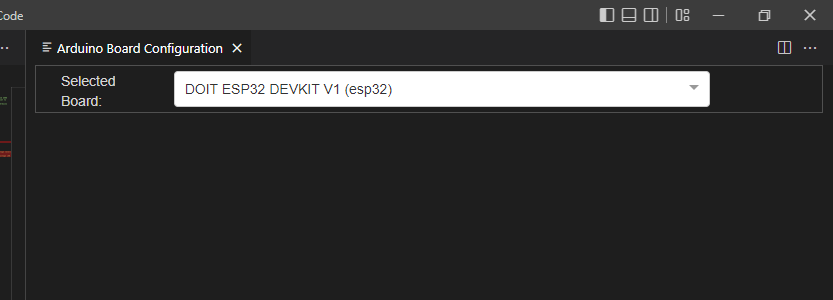
****

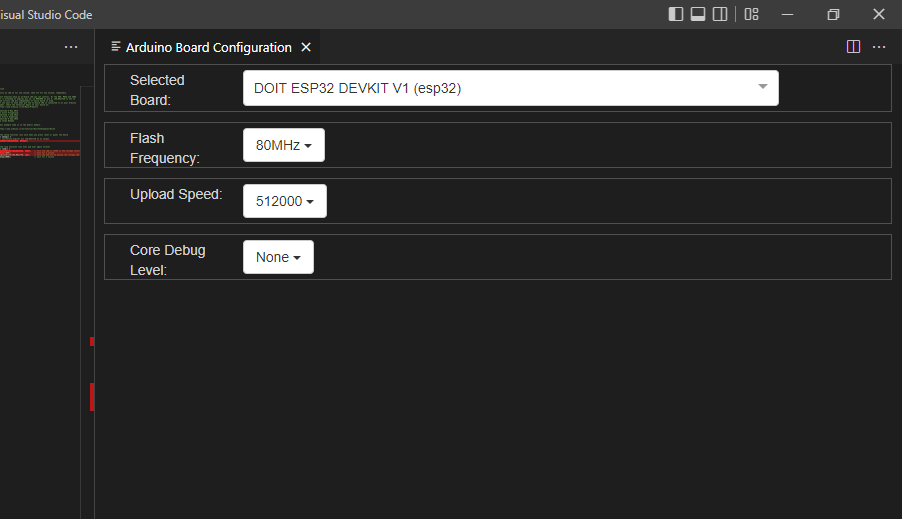




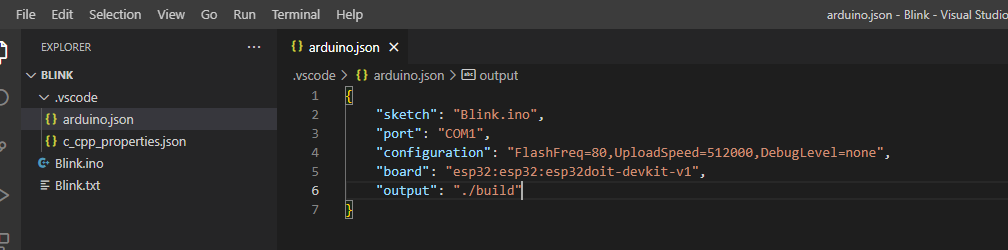
* + Config board type for Blink project



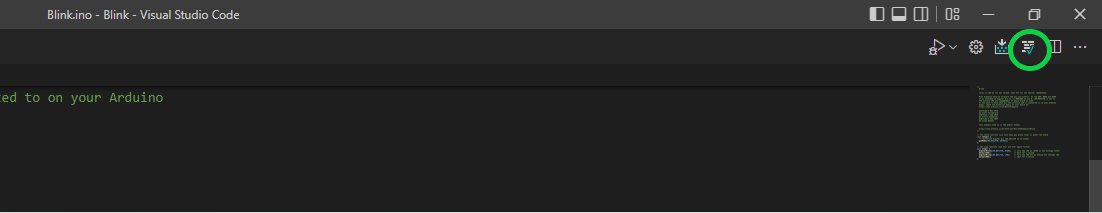


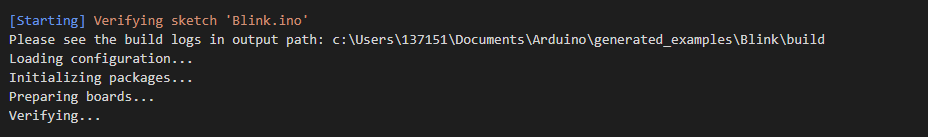


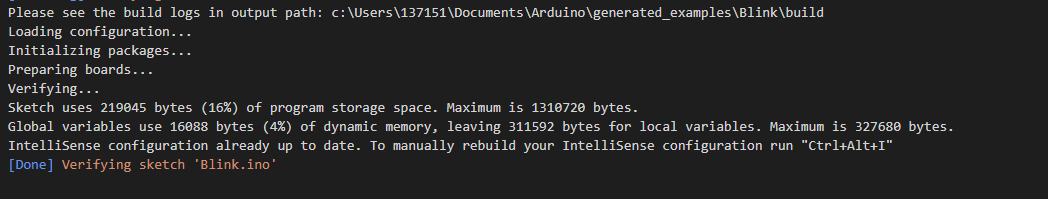
* + Config output directory for build output data

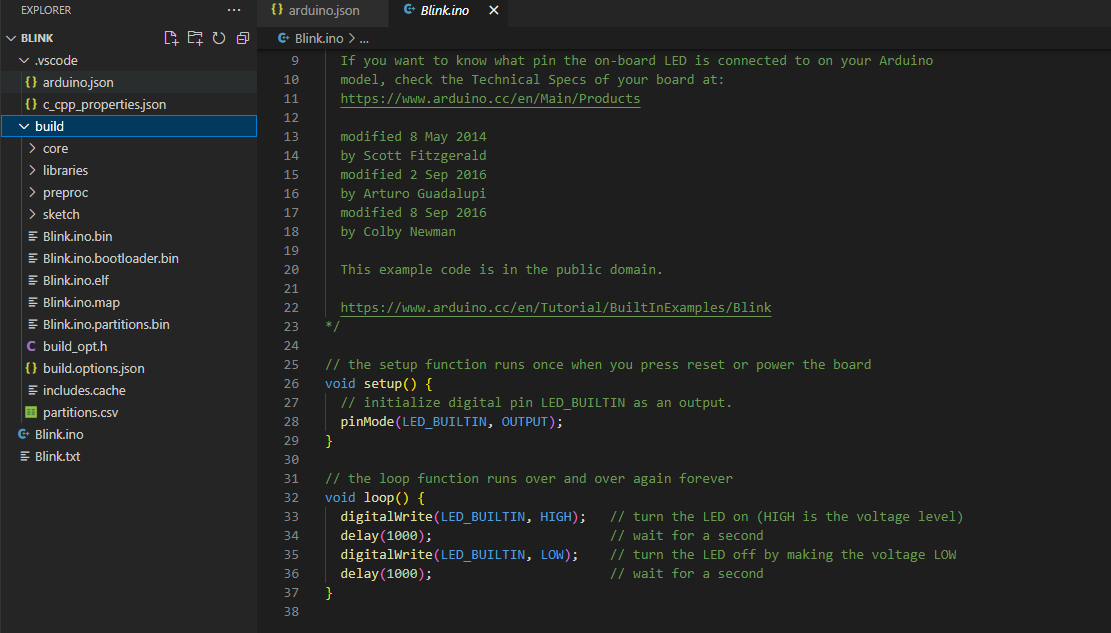


* + Build Blink project

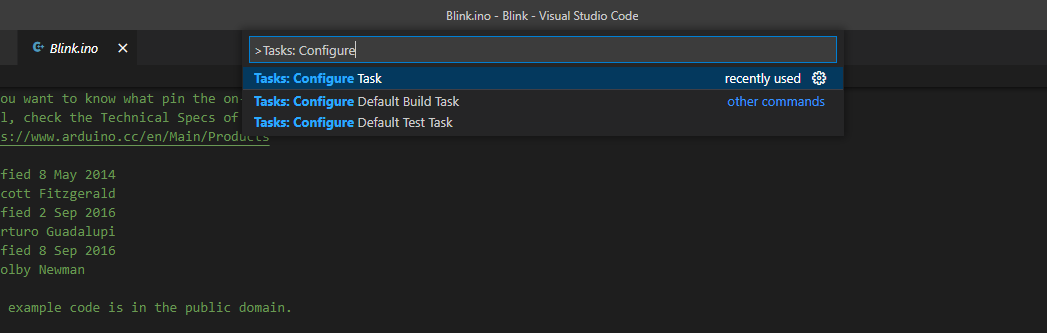




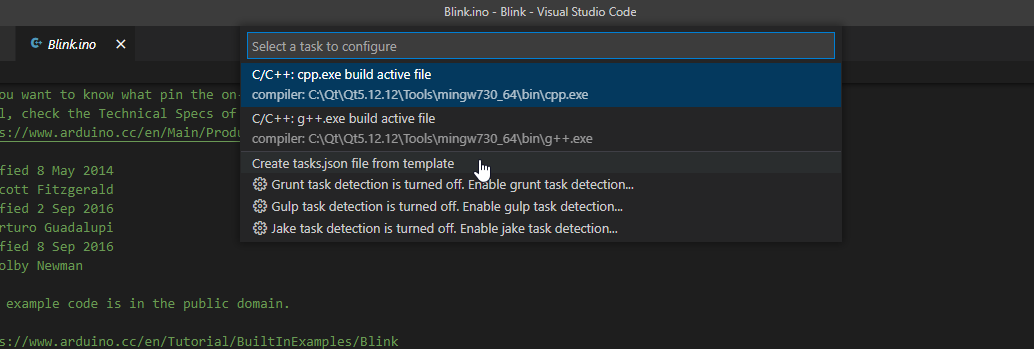




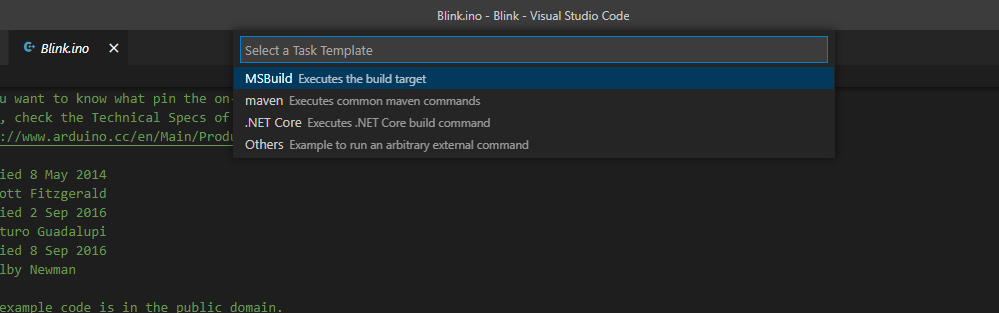
* + Create OpenOCD Server task
    - F1 => Typing “Tasks: Configure”

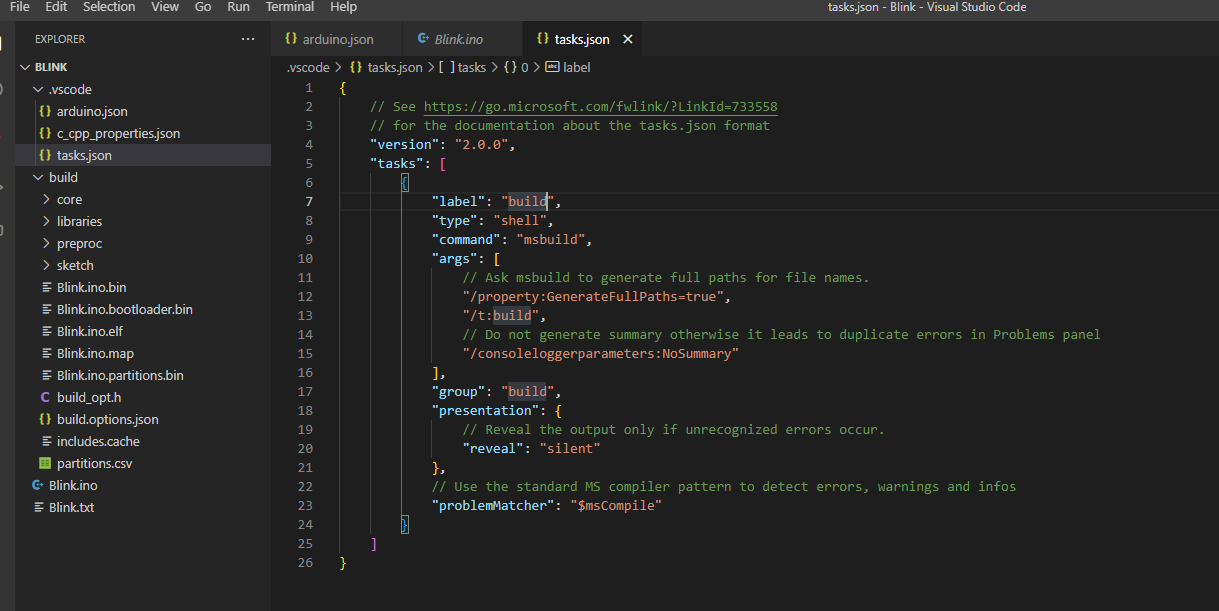
****

* + - Select “Tasks: Configure Task”
    - Select “Create task.json file from template”

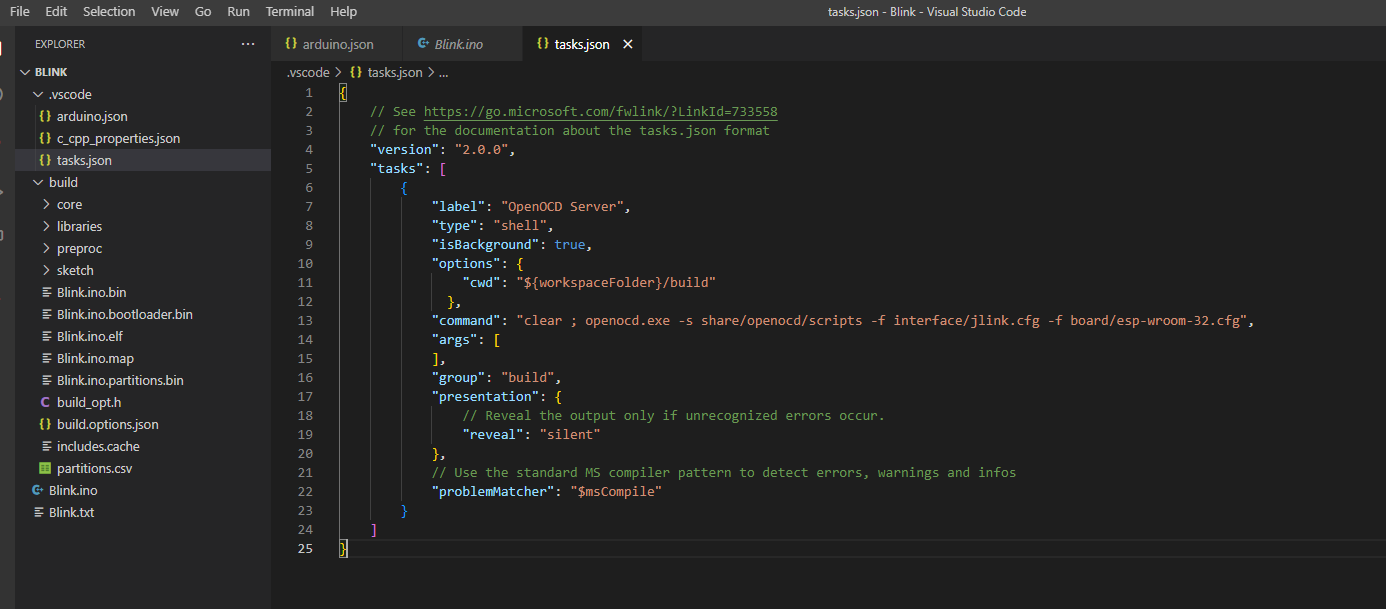
****

* + - Select “MSBuild Executes the build target”

****



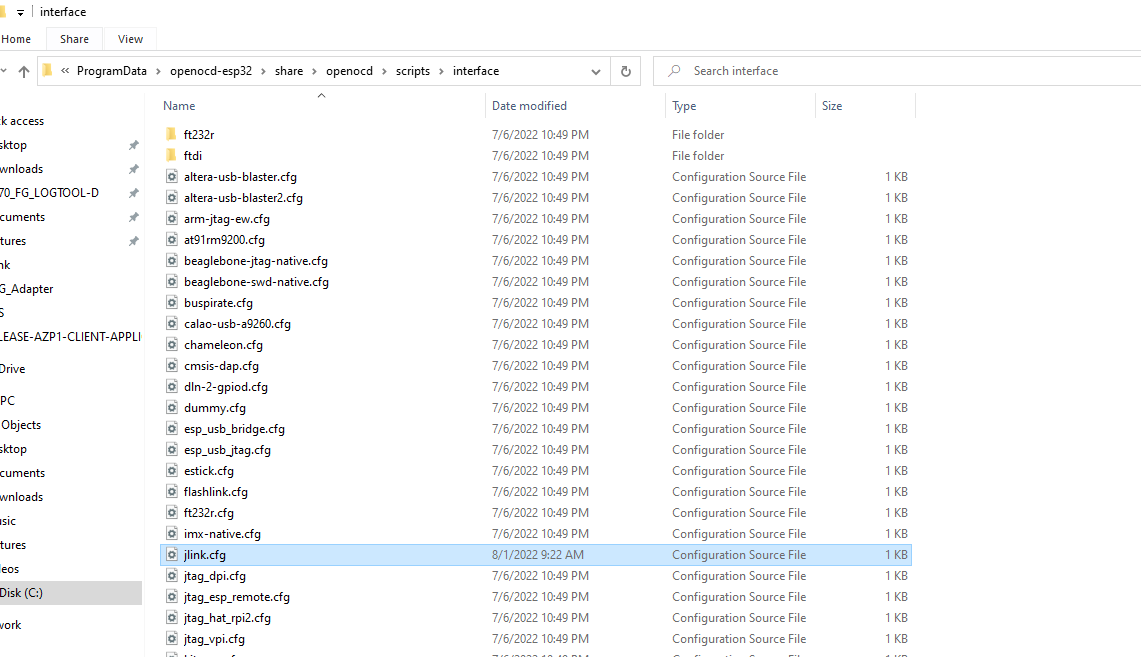
* + - Edit task



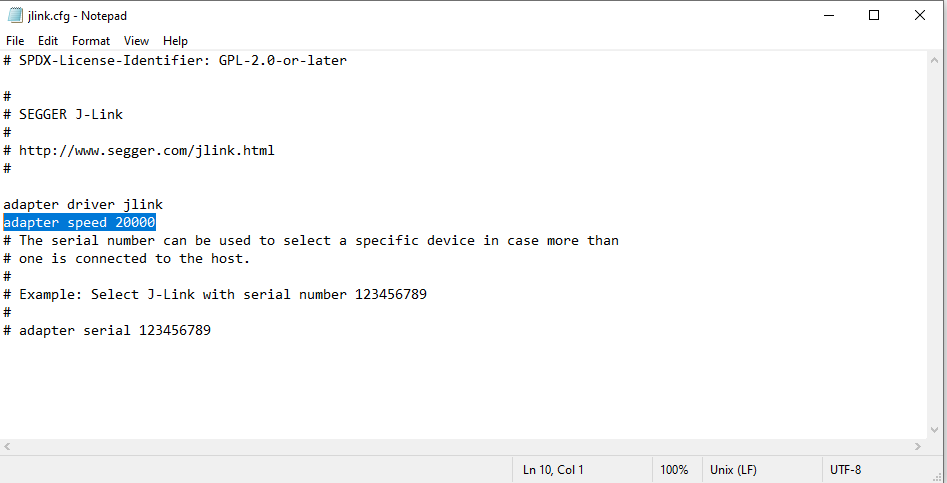
* + - task.json reference content

|  |
| --- |
| * { * // See https://go.microsoft.com/fwlink/?LinkId=733558 * // for the documentation about the tasks.json format * "version": "2.0.0", * "tasks": [ * { * "label": "OpenOCD Server", * "type": "shell", * "isBackground": true, * "options": { * "cwd": "${workspaceFolder}/build" * }, * "command": "clear ; openocd.exe -s share/openocd/scripts -f interface/jlink.cfg -f board/esp-wroom-32.cfg", * "args": [ * ], * "group": "build", * "presentation": { * // Reveal the output only if unrecognized errors occur. * "reveal": "silent" * }, * // Use the standard MS compiler pattern to detect errors, warnings and infos * "problemMatcher": "$msCompile" * }, * { * "label": "Flash Image", * "type": "shell", * "options": { * "cwd": "${workspaceFolder}/build" * }, * "command":[ * "clear ; cd ${workspaceFolder}/build ; openocd.exe -s share/openocd/scripts -f interface/jlink.cfg -f board/esp-wroom-32.cfg -c 'program\_esp ./Blink.ino.bin 0x10000 verify reset exit'" * ], * "args": [ * ], * "group": "build", * "presentation": { * // Reveal the output only if unrecognized errors occur. * // "reveal": "silent", * "close": true// Close terminal when task command execute complete * }, * // Use the standard MS compiler pattern to detect errors, warnings and infos * "problemMatcher": "$msCompile" * }, * { * "label": "Terminate Task", * "type": "shell", * "options": { * "cwd": "${workspaceFolder}/build" * }, * "command":[ * "${command:workbench.action.tasks.terminate}", * "${command:workbench.action.acceptSelectedQuickOpenItem}", * ], * "args": [ * ], * "group": "build", * "presentation": { * // Reveal the output only if unrecognized errors occur. * // "reveal": "silent", * "close": true * }, * // Use the standard MS compiler pattern to detect errors, warnings and infos * "problemMatcher": "$msCompile" * }, * { * "label": "Exit OpenOCD", * "type": "shell", * "options": { * "cwd": "${workspaceFolder}/build" * }, * "command": "echo ${input:terminate}", * "args": [ * ], * "group": "build", * "presentation": { * // Reveal the output only if unrecognized errors occur. * // "reveal": "silent", * "close": true * }, * // Use the standard MS compiler pattern to detect errors, warnings and infos * "problemMatcher": "$msCompile" * }, * ], * "inputs": [ * { * "id": "terminate", * "type": "command", * "command": "workbench.action.tasks.terminate", * "args": "OpenOCD Server" * } * ] * } |

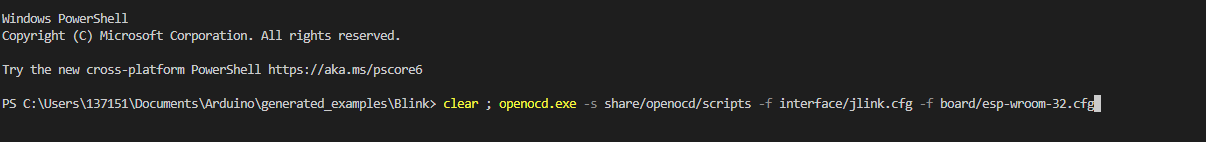
* + - Edit jlink.cfg file

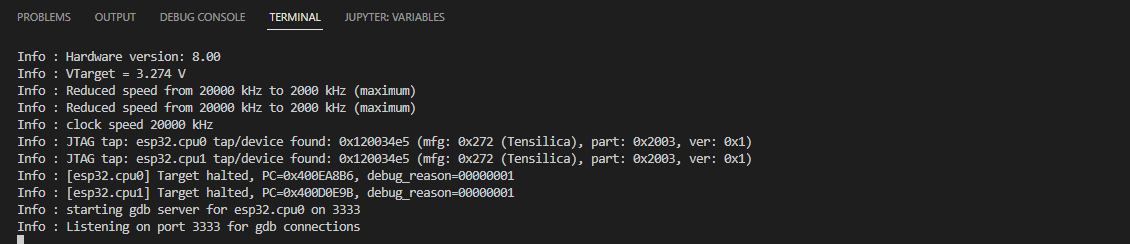


* + - * Add adapter speed parameter

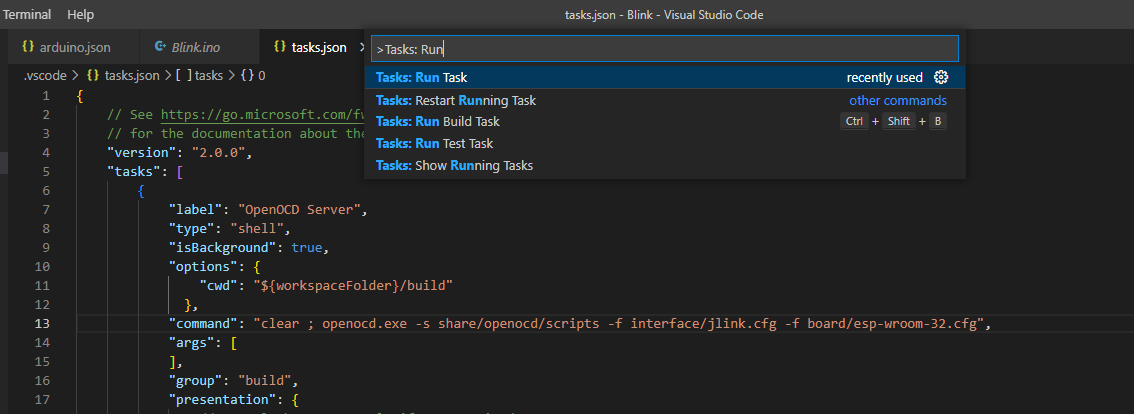


* + - Check command of task

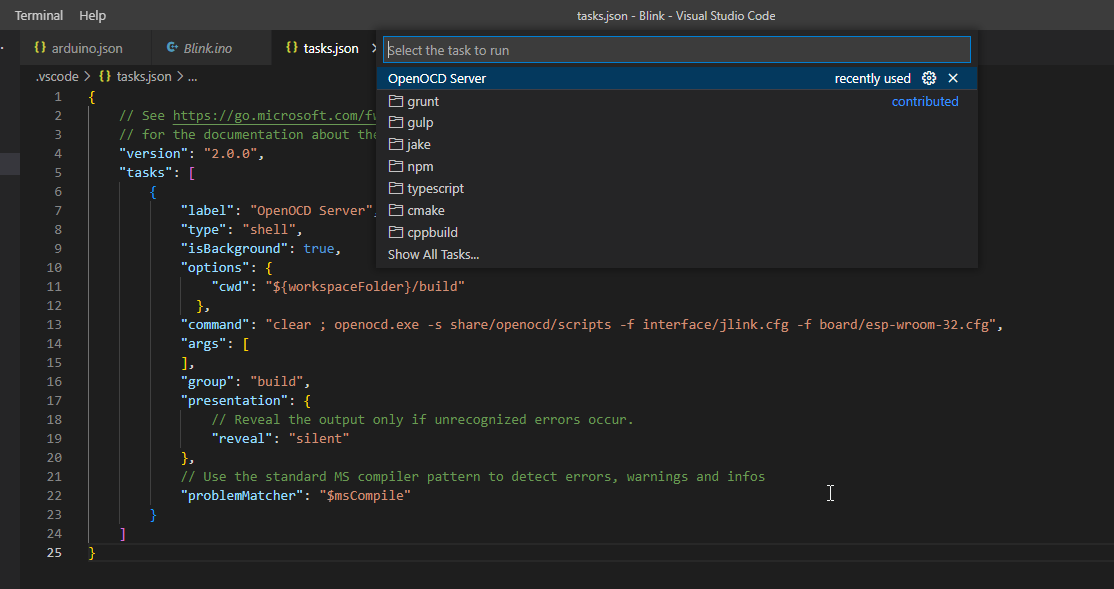


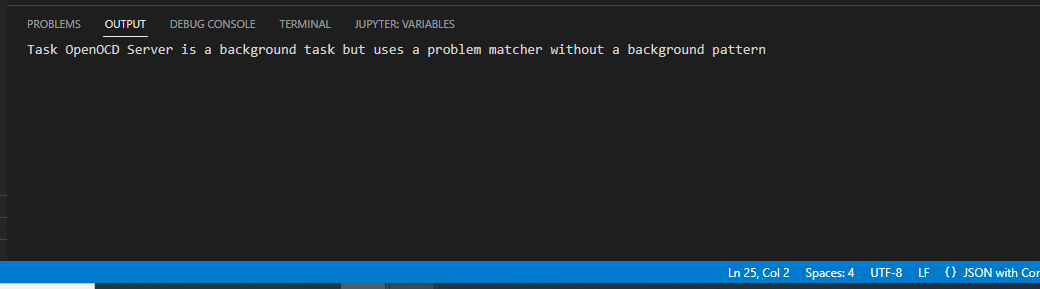


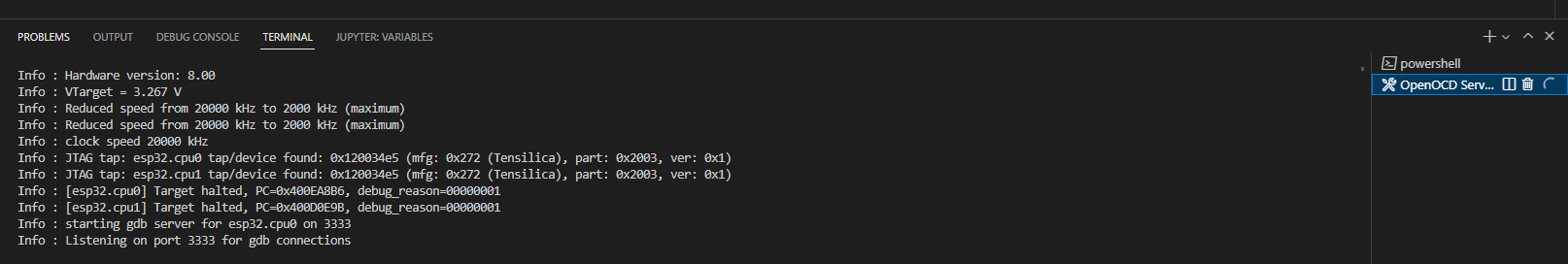
* + - Check task by run task from Visual Studio Code
      * F1 => Typing “Tasks: Run”

****

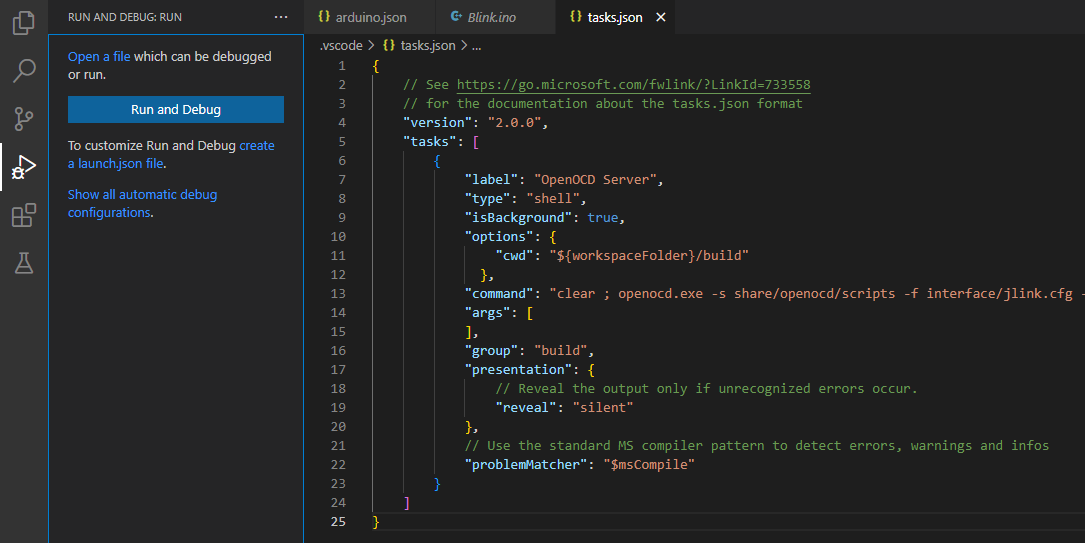
* + - * Select “Tasks: Run Task” => Select “OpenOCD Server” task

****

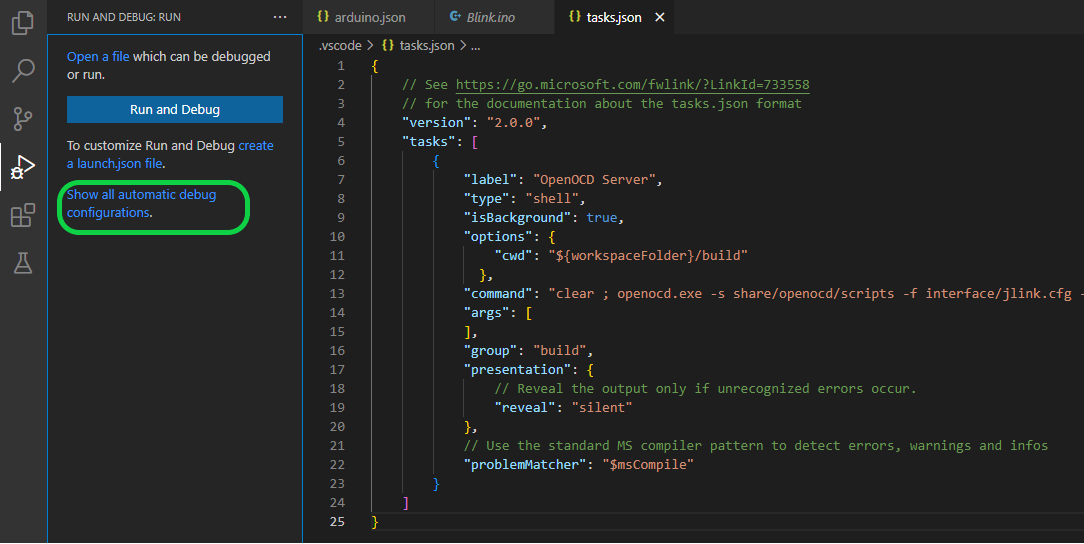




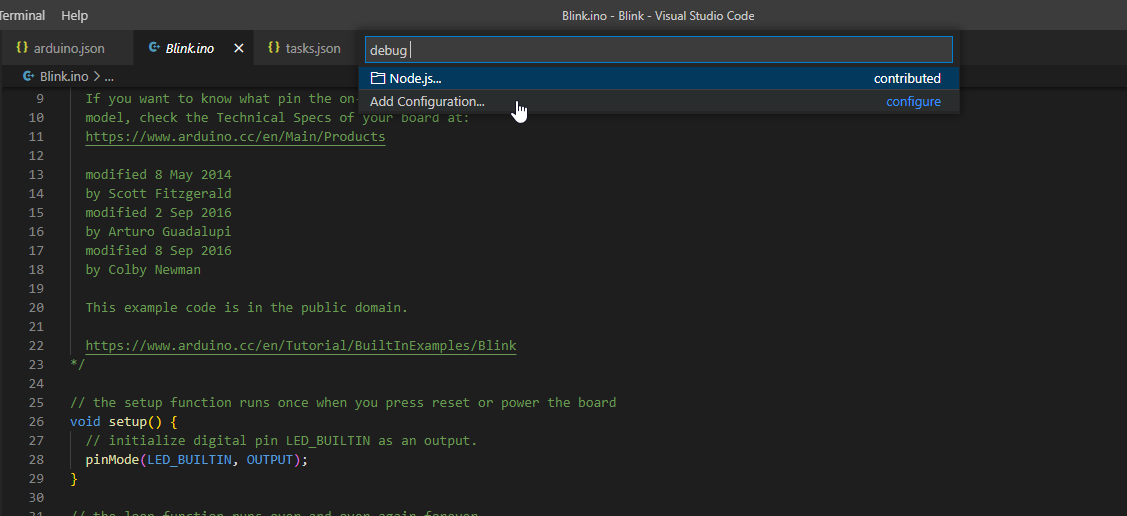
* + Create “gdb Launch”
    - Press “Ctrl + Shift + D”



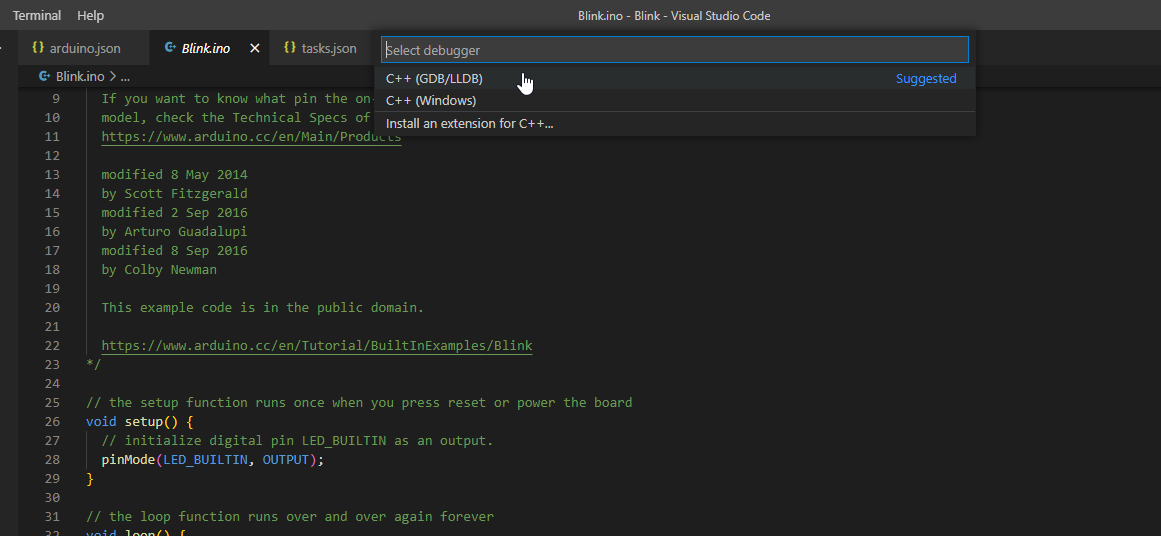
* + - Select “Show all automatic debug configurations”

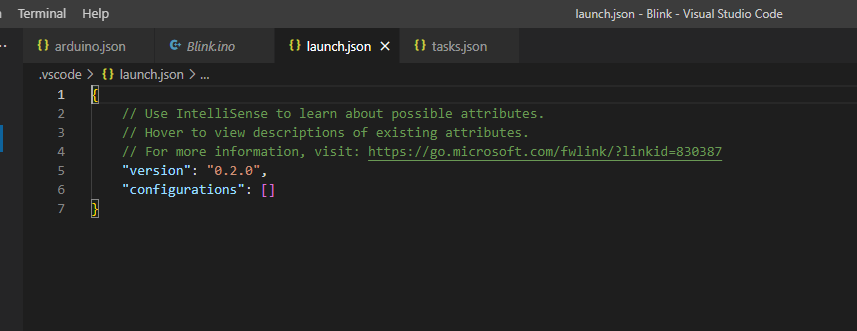


* + - Select “Add configuration…”

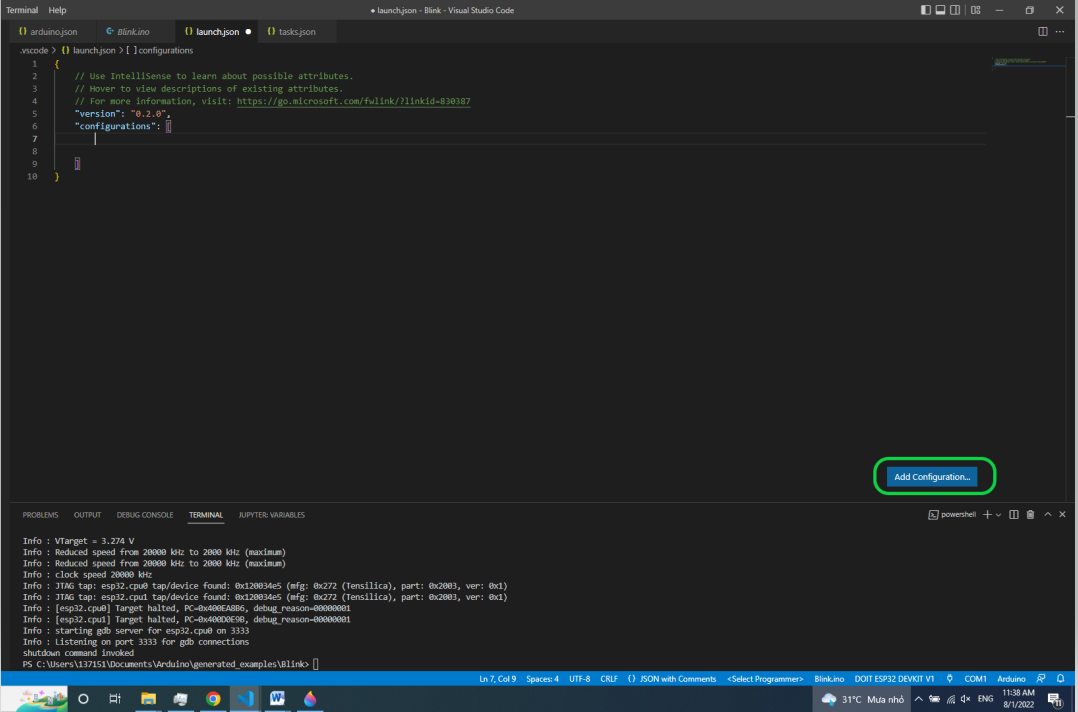
****

* + - Select “C++ (GDB/LLDB)”

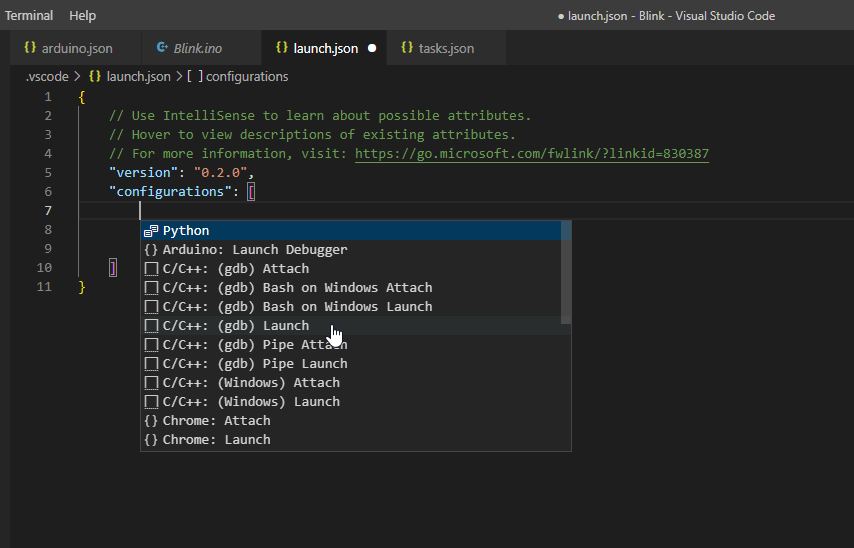
****

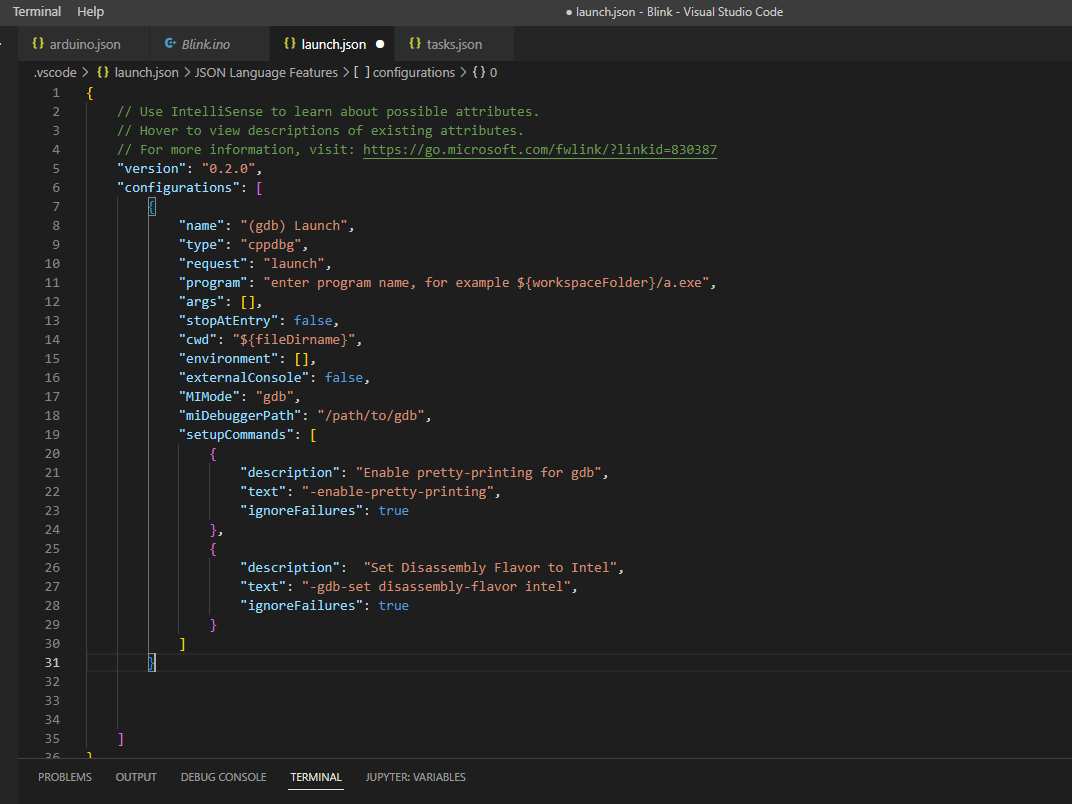


* + - Select “Add Configuration”

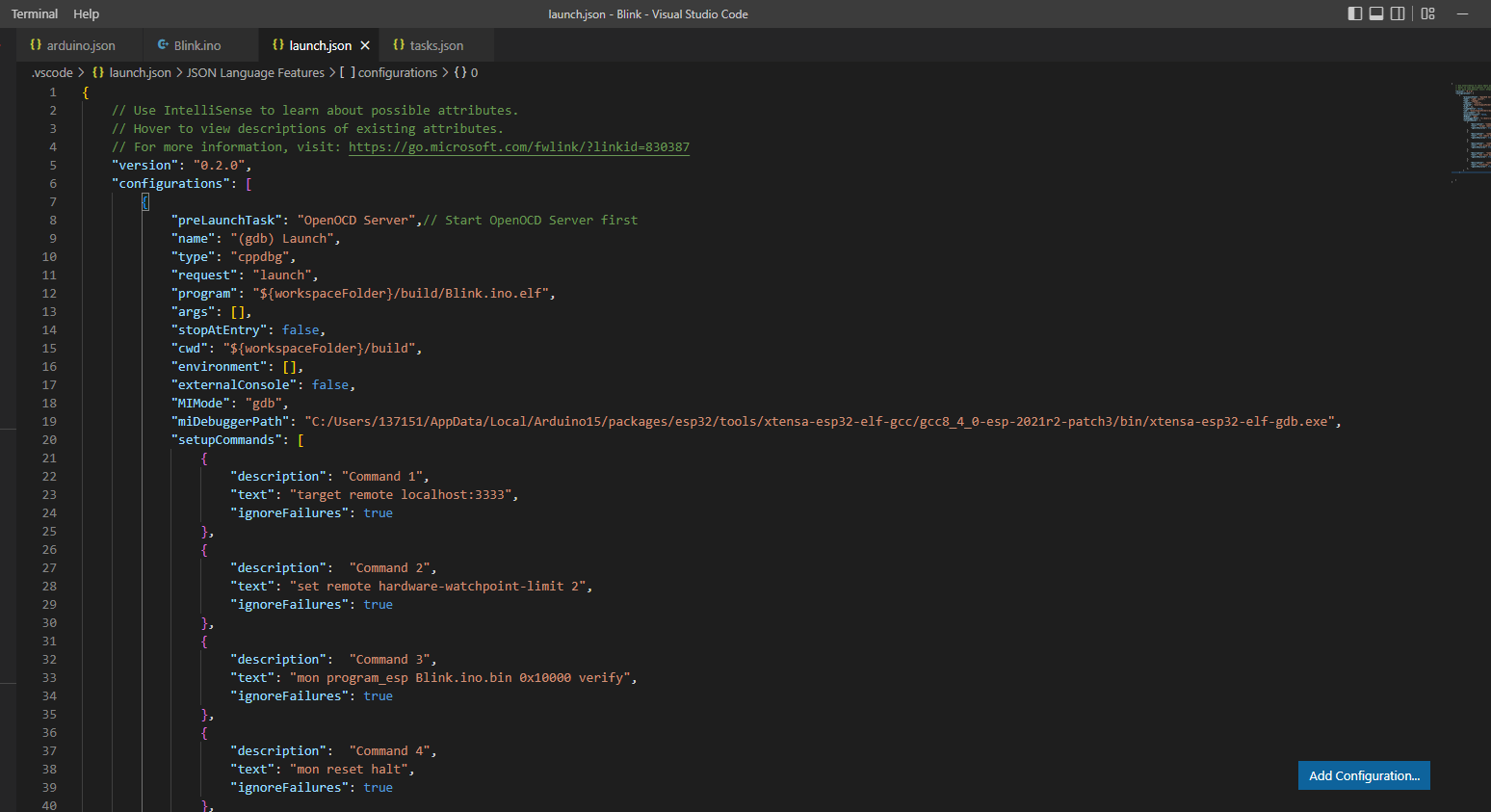


* + - Select “C/C++: (gdb) Launch”

****



* + - Edit config

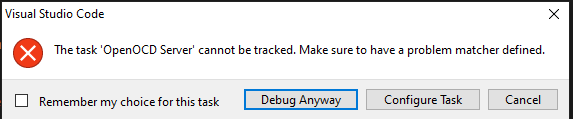


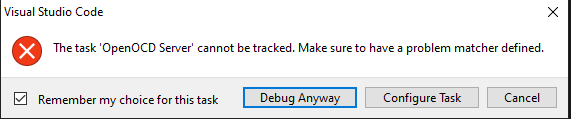


* + - launch.json reference content

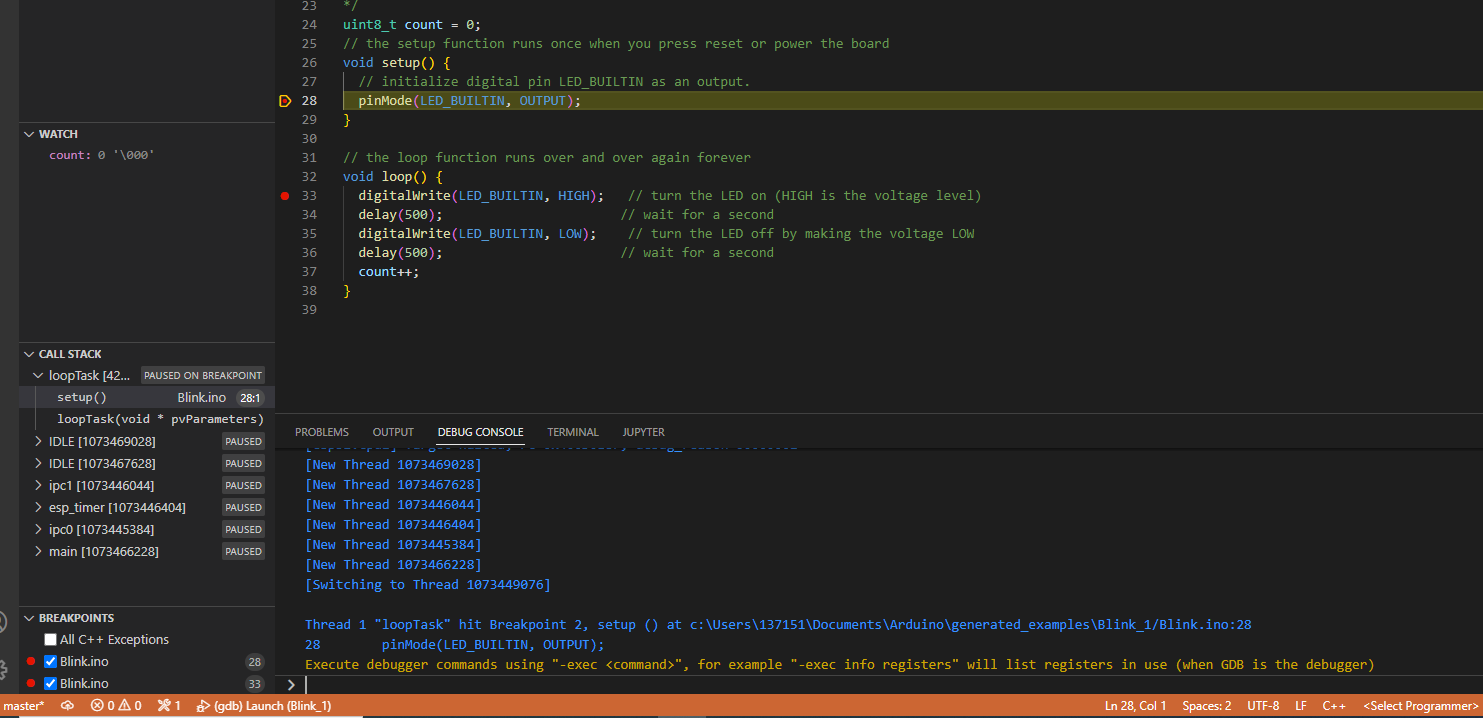
|  |
| --- |
| * { * // Use IntelliSense to learn about possible attributes. * // Hover to view descriptions of existing attributes. * // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387 * "version": "0.2.0", * "configurations": [ * { * "preLaunchTask": "OpenOCD Server",// Start OpenOCD Server first * "name": "(gdb) Launch", * "type": "cppdbg", * "request": "launch", * "program": "${workspaceFolder}/build/Blink.ino.elf", * "args": [], * "stopAtEntry": false, * "cwd": "${workspaceFolder}/build", * "environment": [], * "externalConsole": false, * "MIMode": "gdb", * "miDebuggerPath": "C:/Users/137151/AppData/Local/Arduino15/packages/esp32/tools/xtensa-esp32-elf-gcc/gcc8\_4\_0-esp-2021r2-patch3/bin/xtensa-esp32-elf-gdb.exe", * // Connect to OpenOCD server * "setupCommands": [ * { * "description": "Enable pretty-printing for gdb", * "text": "-enable-pretty-printing", * "ignoreFailures": true * }, * { * "description": "Command 1", * "text": "target remote localhost:3333", * "ignoreFailures": true * }, * ], * // Seperate with setupCommands to fix issue "Info : dropped 'gdb' connection" * "postRemoteConnectCommands": [ * { * "description": "Respect Hardware Limitations (as prescribed by Espressif)", * "text": "set remote hardware-watchpoint-limit 2", * "ignoreFailures": false * }, * { * "description": "Hard Reset and Immediately Halt", * "text": "monitor reset halt", * "ignoreFailures": true * }, * { * "description": "Flush Internal Register Cache", * "text": "flushregs", * "ignoreFailures": false * }, * // { * //  "description": "Set Temporary Hardware Assisted Breakpoint at `app\_main`", * //  "text": "thbreak setup", * //  "ignoreFailures": false * // }, * // { * //  "description": "Shutdown GDB Server on GDB Detach", * //  "text": "monitor [target current] configure -event gdb-detach { shutdown }", * //  "ignoreFailures": true * // }, * { * "description": "Reset MCU when stop debug session", * "text": "monitor [target current] configure -event gdb-detach { reset }", * "ignoreFailures": true * }, * ], * "launchCompleteCommand":"None",// To fix issues [ERROR: Unable to start debugging. Unexpected GDB output from command "-exec-run". Don't know how to run.  Try "help target".] * // "internalConsoleOptions": "openOnSessionStart", * // "logging": { "engineLogging": true },// Enable log of GDB commands * "postDebugTask": "Exit OpenOCD" * } * ] * } |

* + Start Debug
    - Run => Start Debugging
    - If error: The task ‘OpenOCD Server’ cannot be tracked. Select Debug Anyway.





* + - If fail try to 3 times it will worked.
  + Log output from GDB server



* **Debug flow:**
  + Build project
  + Flash firmware to ESP32
    - F1 => Tasks: Run Task => Flash Image
  + Start debug
    - F5
  + Stop debug. Don’t restart.

**4.**